Construction Research and Application of Poverty Alleviation Knowledge Graph

Hongyan Yun[1], Ying He[2](✉), Li Lin[1], Zhenkuan Pan[1],
and Xiuhua Zhang[1]

[1] College of Computer Science & Technology, Qingdao University,
Qingdao 266071, Shandong, China
[2] School of Electronic Information, Qingdao University, Qingdao 266071,
Shandong, China
`Yunhy2001@163.com`

Abstract. Based on the integration of multi-source data, an approach of domain-specific knowledge graph construction is proposed to guide the construction of a "people-centered" poverty alleviation knowledge graph, and to achieve cross-functional and cross-regional sharing and integration of national basic data resources and public services. Focusing on "precise governance and benefit people service", poverty alleviation ontology is constructed to solve semantic heterogeneity in multiple data sources integration, and provide an upper data schema for poverty alleviation knowledge graph construction. Karma modeling is used to implement semantic mapping between ontology concepts and data, and integrate multi-source heterogeneous data into RDF data. The RDF2Neo4j interpreter is developed to parse RDF data and store RDF data schema based on the graph database Neo4j. Based on visualization technology and natural language processing technology, Poverty Alleviation Knowledge Graph Application System is designed to achieve knowledge graph query and knowledge question answering function, which improved the application value of government data.

Keywords: An approach of knowledge graph construction · Poverty alleviation knowledge graph · Neo4j graph storage · Bayesian classification · Knowledge question answering

1 Introduction

There is a huge amount data spread across the Web and stored in databases that we can use to build knowledge graph. However exploiting this data to build knowledge graph is difficult due to the heterogeneity of the sources. Knowledge graph is a semantic network and a data structure that can store knowledge, and infer new knowledge to users. Knowledge graph organizes massive Internet data into knowledge network through deep semantic analysis and data mining, searches and displays knowledge intuitively. Knowledge graph has strong knowledge organization ability and semantic processing ability, which provides important guarantee for Internet intelligent search, big data analysis, intelligent question and answer, personalized recommendation and so on [1].

W. Ni et al. (Eds.): WISA 2019, LNCS 11817, pp. 430–442, 2019.
https://doi.org/10.1007/978-3-030-30952-7_42

The data source of general knowledge graph is mainly Internet Web pages, which identify entity data from Web pages and then summarize data patterns. For example, DBpedia uses structure data in Wikipedia entries as data sources, supports 125 languages and covers more than 10 million entities [2]; Zhishi.me uses structured data in Baidu Encyclopedia, Interactive Encyclopedia and Wikipedia as data sources, extracts entities and integrates entity information by structured information extraction method to build a Chinese general knowledge graph [3]. The Knowledge Workshop Laboratory of Fudan University developed and maintained CN-DBpedia, which is a large-scale open Encyclopedia Chinese knowledge graph covering tens of millions of entities and hundreds of millions of relationships. Domain knowledge graph focuses on entities and concepts of knowledge in specific fields or industries, such as agricultural knowledge graph, traditional Chinese medicine knowledge graph, financial knowledge graph, legal knowledge graph, etc. Domain knowledge graph can obtain complete knowledge for analysis and solution of domain problems.

At present, there is a lack of knowledge graph in the field of government big data application and the basic data of many government departments only exists in isolation, which fails to realize the interconnection and interoperability of data. Aiming at the lack of government poverty-alleviation knowledge graph, meeting the needs of precise governance and benefiting the people, this paper builds a “people-centered” poverty-alleviation knowledge graph by integrating national basic information data such as population, corporate enterprises and data resources in the fields of credit and social security. It will make efficient use of open government data on poverty alleviation, integrate isolated data nodes into knowledge base, and provide users with a friendly information service platform for poverty alleviation.

From the perspective of rapid integration of multi-source heterogeneous data, this paper proposes an approach to construct knowledge graph. A crawler system is constructed to obtain data and information related to poverty alleviation from local government poverty-alleviation websites. Fuse data from the poverty alleviation database published by the government to construct the poverty alleviation ontology that provides the upper data model for the construction of poverty-alleviation knowledge graph. Neo4j storage scheme for triple data is developed for structured and semi-structured. Graph storage method effectively improves the efficiency and scalability of data retrieval, and provides an important storage guarantee for further expansion and maintenance of knowledge graph. Based on entity recognition and Bayesian classification, an application system is designed to implement knowledge graph query and intelligent question-answer functions. This application system improves the application value of open government data.

2 An Approach of Building Knowledge Graph

The general knowledge graph covers the knowledge field and the data, and the construction technology covers many fields such as semantic technology, natural semantic processing, machine learning, deep learning, etc. The construction process is complex and the construction cycle is long [4]. Domain knowledge graph covers relatively less domain-specific knowledge and data, which shortens the construction cycle.

We propose an approach to build knowledge graph. We present the techniques using poverty alleviation domain as an example, though the general approach can be applied to other domains. Figure 1 shows the architecture of overall approach to construct knowledge graph for poverty alleviation, which can meet the needs of effective utilization and accurate poverty alleviation of government data in the field of poverty alleviation. The construction process of knowledge graph includes data acquisition, knowledge fusion and knowledge processing.

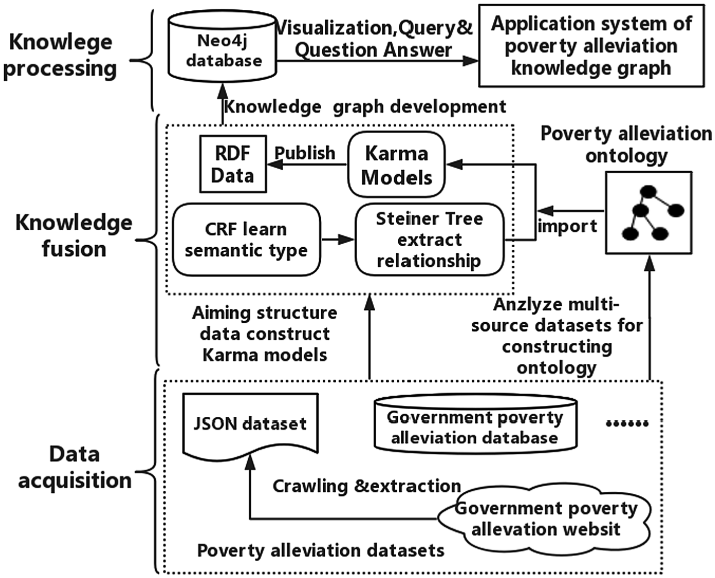

Knowlege processing
Neo4j database
Visualization,Query& Question Answer
Application system of poverty alleviation knowledge graph
Knowledge graph development
Knowledge fusion
RDF Data
Publish
Karma Models
Poverty alleviation ontology
import
CRF learn semantic type
Steiner Tree extract relationship
Aiming structure data construct Karma models
Anzlyze multi-source datasets for constructing ontology
Data acquisition
JSON dataset
Government poverty alleviation database
••••••
Crawling &extraction
Government poverty allevation websit
Poverty alleviation datasets

Fig. 1. An approach of Poverty-alleviation knowledge graph construction.

2.1 Data Acquisition

The data in the field of poverty alleviation is mainly composed of two parts: one is the structured data provided by government departments stored in MySQL database, the other requires finding relevant municipal governmental poverty alleviation websites and extracting the required information from those pages.

Based on Java crawler framework named WebMagic (https://gitee.com/flashsword20/webmagic), a crawler system is designed to support crawling at scale. The crawler system mainly includes downloader, page parser, scheduler and pipe component. The four elements are organized in the Spider container, and regular expressions are written to crawl the required website data through interactive and process execution. The workflow of this crawler system includes the following five steps. ① The initial URL of the reptile is the Shandong Provincial Poverty Alleviation Supply and Demand Platform (http://ax.sdfp.gov.cn/) and Chongqing Poverty Alleviation Open Website (http://www.cqfp.gov.cn/channels). ② The downloader uses Apache HttpClient as the download component to initiate a request for the initial URL to obtain the Web page object Page. ③ Page parser uses process() method in Processor to parse web pages, uses Jsoup to parse HTML pages into DOM trees, extracts useful information resources and discovers new seed URLs through CSS Selector. Aiming at government's poverty alleviation website, take the poor householders as the center to crawl the structured information in the website. ④ The scheduler manages the URLs to be fetched and delete-duplicate operations. ⑤ Pipeline is responsible for the processing of extraction results, mainly including saving data to files or databases.

After data extraction, we convert pages related to poor households into structured knowledge for storage. The extracted relational instances are defined in the following forms: <conceptual instance1, relational concepts, conceptual instance2>. The basic information of the poor households can be expressed in the form of multiple triples: <31185489099, reason of poverty, "illness">, <31185489099, detailed address, "Wenshangji Town, Chengwu County, Heze City>, <31185489099, helper, "Yuan Cunkui"> … etc.

2.2 Knowledge Fusion

Knowledge fusion solves the problem of data normalization and multi-source heterogeneous data integration. Construct domain ontology and Karma models (http://isi.edu/integration/karma/) to quickly integrate multi-source heterogeneous data. The integration scheme of multi-source heterogeneous data based on ontology and Karma modeling has been validated effectively in the fields of food security [5] and regional armed conflict event data (ACLED) integration [6]. Karma is an ontology-based data integration tool, which supports to import multiple data sources, including spreadsheets, relational databases, XML, CSV, JSON, etc. Through data cleaning and standardization, Karma builds a model or semantic description for each data source to integrate data across multiple data sources. Karma uses conditional Random Field (CRF) model to learn and recognize semantic types to complete mapping between data and ontology concepts. Under the constraint of ontology, Karma uses Steiner Tree algorithm to extract the relationship between data in data sources. Users can interact with the system to adjust the automatically generated model. In this process, users can convert data format according to their needs, standardize or reconstruct the data expressed in different formats. In order to support rapid modeling of big data, small batch data can be imported to build Karma model. Karma supports to export R2RML (RDB to RDF Mapping Language) model and then directly apply the model to the complete big data set [7], which can be published into a unified RDF data, so as to achieve the purpose of rapid modeling. Poverty alleviation ontology constructing will be described in Sect. 3. Karma modeling will not be described in this paper.

2.3 Knowledge Processing

This paper builds an interpreter based on the RDF data in poverty alleviation field, and stores RDF data into Neo4j graph database (introduced in Sect. 4 of this paper), so as to improve the expansibility, maintainability and the ability of visual analysis and processing of the knowledge graph. This paper will describe design and implementation of application system of poverty-alleviation knowledge graph in Sect. 5.

3 Poverty Alleviation Ontology

Ontology formally describes the semantic relationship between concepts in the domain, which provides a unified perspective for treating domain knowledge and solving domain problems [8]. Domain ontology can provide the upper data model for knowledge graph, and it is an important part of knowledge graph. By analyzing poverty alleviation data sets of different municipal government departments, main class, data property and object property are extracted to construct poverty alleviation ontology. Some extracted triples are shown as follows. By analyzing extracted triples, define main classes and properties of poverty alleviation ontology are shown in Table 1.

Table 1. Defined main classes and properties of Poverty-alleviation ontology.

Class/Attribute name	Class/Property	Meaning
Helper	Class	Helper people
Poor households	Class	Poor households
Family	Class	Poor family
GovHelp	Class	Support project
has_helper	ObjectProperty	
has_govhelp	ObjectProperty	
has_family	ObjectProperty	
has_name	DataProperty	Name of poor household
has_id	DataProperty	ID of poor household
has_address	DataProperty	Address of poor household
has_phone	DataProperty	Phone of poor household
has_starttime	DataProperty	Start time of support
has_endtime	DataProperty	End time of support
has_proname	DataProperty	Name of support project
has_protype	DataProperty	Type of support project

{<poor household, has a family, family information>; <poor household, has a householder, name>; <poor household, has family address, address information>; <poor household, has phone, phone number>; <poor household, has a helper, helper name>; <poor household, has a support project, support project name>}

{<helper, start time to support, start time> ;<helper, end time to support, end time>}

{<GovHelp, has a name of support project, name of support project>; <GovHelp, has a type of support project, type of support project>; <GovHelp, has information of support project, information of support project>}

{<family, has family introduce, introduce information>; <family, has a family ID, family ID>; <family, has number of students, number>; <family, has information of annual income, annual income>; <family, has house information, house information>}

Gradually refine concepts and its relationship, construct poverty alleviation ontology is shown in Fig. 2. Data property in the graph is identified by dotted lines with arrows, and object property is identified by solid lines with arrows.

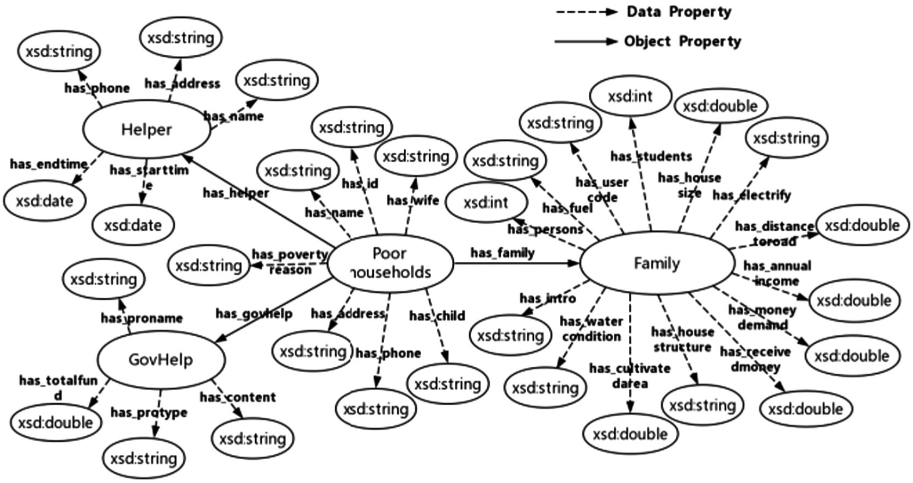


Fig. 2. Poverty alleviation ontology.

4 Knowledge Graph Building and Storage

Poverty alleviation ontology defines the terminology for representing nodes and edges of the knowledge graph. The key decision in the design of knowledge graph is the desire to record provenance data for each node and edge of the graph [9]. Neo4j is one of the current mainstream graph databases and has the characteristics of high performance, high reliability, scalability and transaction support. Neo4j follows the data model of attribute graph, which solves the problem of performance degradation when a large number of connection queries are made in relational databases. The graph-based traversal algorithm enables it to traverse nodes and edges at the same speed. Its traversal speed has nothing to do with the magnitude of graphics, so it can also perform well in big data.

In the view of the complexity and dynamics of data relations and the extension and maintenance of knowledge graph, this paper uses Neo4j graph database to store knowledge graph. The representation of knowledge graph is a unified set of triples. Poverty-alleviation knowledge graph is imported into Neo4j as shown in Fig. 3.

The RDF2Neo4j interpreter is designed to import structured data into Neo4j graph database. The specific steps are as follows: ① Export multiple tables from MySQL database into CSV format. ② Using Cypher statement to load several database tables, each table is regarded as a kind of data type, each record as different instance data of this type, encapsulating corresponding attributes and attribute values according to requirements, creating nodes in the knowledge graph. ③ According to the foreign key

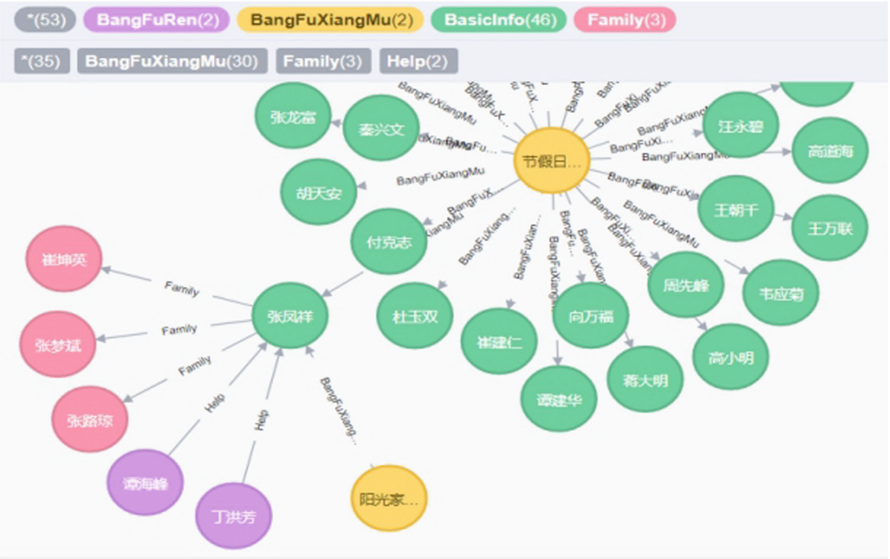


Fig. 3. Poverty-alleviation knowledge graph in Neo4j.

association between different tables, it maps to the relation type of different types of nodes, and creates relations in the knowledge graph.

5 Poverty Alleviation Knowledge Graph Application System

Data visualization technology presents data intuitively and interactively. Combine knowledge graph with visualization can accurately analyze and display data [10]. The poverty-alleviation knowledge graph application system has four-layer architecture, including data layer, data query layer, business logic layer and presentation layer, as shown in Fig. 4. The data layer stores poverty alleviation data in Neo4j graph database. Data query layer mainly uses Spring Boot framework to build micro services, provides RESTful interface, uses Spring Data module to operate Neo4j graph database, and writes Cypher statements to interact with Neo4j graph database. Business logic layer mainly deals with data, and further encapsulates returned data by calling data query layer to complete data statistical analysis and format specification. The business logic is compiled according to functions that need to be completed, and the encapsulated data is transmitted to the presentation layer. The data exchange format is JSON. Presentation layer passes received data to the front-end for rendering, and uses Echarts components and HTML5 to visualize front-end pages of the system. The application system of poverty alleviation knowledge graph has designed and implemented functions including display, query (inquiry based on the names of poor households, projects and relationships among poor households) and intelligent knowledge question and answer for users. This section elaborates on development of knowledge graph query and question-and-answer function.

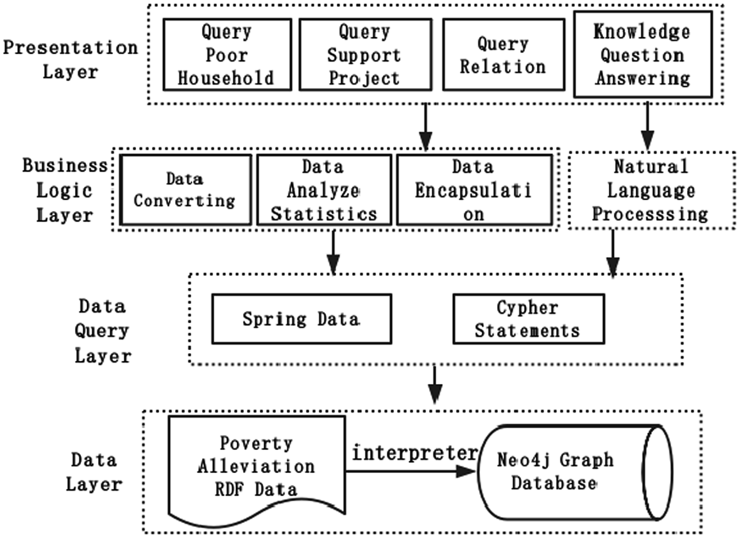


Fig. 4. P architecture of Poverty-alleviation knowledge graph application system.

5.1 Query Names of Poor Households and Support Projects

Users input the name of the poor household head and click on Query button to display a one-dimensional relationship. Different entity types are identified by different colors nodes, and different relation types are identified by text display and direction identification. The main types of relationships are support projects, help people, family information and so on. By using support project as the keyword to query knowledge graph, the multi-dimensional relationship of support project entity is returned, which can clearly show the relationship between different poor households. The steps to implement the query function are as follows.

Use ECharts to Build the Oriented Layout. Echarts is a powerful, interactive charting and visualization library offering an easy way to add intuitive, interactive and highly customizable charts to your need (http://echarts.apache.org/). ECharts defines a variety of icon styles and functions. According to users' actual needs, developers use corresponding JS code to send requests to the background and render returned data. In HTML, the ECharts container is created first. The option object is set in the container and the related code of force-oriented graph is introduced. The title, type, icon, node style and callback function of the graph are specified. The main configurations of ECharts used in knowledge graph visualization design are shown in Table 2.

Send Requests. The front-end uses AJAX in JQuery library to send asynchronous requests. It can refresh local data without refreshing web pages, parse and transmit returned data. Obtain query conditions entered by users and send POST requests to the background, call different methods to query, render returned JSON data in a specific format. For example, user query one-dimensional relationship with "a poor household as the center" and multi-dimensional relationship with "a poverty alleviation project as the center".

Table 2. Configuration of Echarts relation graph.

series[i]-graph	Relational graph components
series[i]-graph.force	Boot layout-related configuration items
series[i]-graph.roam	Mouse zooming and panning roaming
series[i]-graph.symbol	Node Styles in Diagrams
series[i]-graph.edgeSymbol	Edge Styles in Diagrams
series[i]-graph.data[i]	Data Format of Specific Relational Graph
series[i]-graph.categories[i]	Types of specific nodes

5.2 Query Relation of Entities

Users input names of two poor householders that need to be queried. If exists a relationship, the shortest path relationship graph between the two poor households will be displayed. Different entity types are identified by different nodes of colors, and different relationship types are identified by text display and direction marking. If there is no relationship, two isolated nodes are displayed.

The system obtains inputted names of two householders and makes a request to the server. The business logic layer of the server receives name for verification. After verification, call the method of data query layer to transfer into parameters. The shortest path function shortestPath() of Neo4j graph database is used to query the path between two poor householders and return result set. The system parses and processes the result set, transfers JSON data encapsulated in a specific format to ECharts component rendering, and finally displays it in graph.

5.3 Knowledge Question Answering Design

Intelligent Knowledge Question Answering (KQA) refers to semantic understanding and parsing of a given natural language question, and then querying the knowledge base to get the answer. For a natural language problem, the challenge is how to map question to knowledge base (corresponding data query statement representation) and convert natural language questions into understandable expressions in databases.

The main processes of question processing include question segmentation, information extraction and text classification. We design a knowledge question answering system as shown in Fig. 5, which consists of constructing question training set, preprocessing natural language questions input by users, including word segmentation, named entity recognition, using Naive Bayesian classification method to classifiers question, putting the results of question text preprocessing into the classifier model, and calculating the short text classification of question sentences. Execute queries corresponding to categories to get answers from knowledge graph queries and implementation process includes the following four steps.

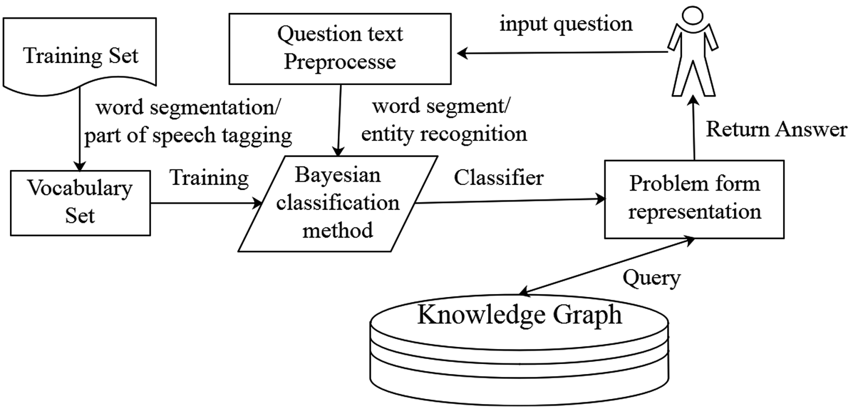

Training Set
Question text Preprocesse
input question
word segmentation/ part of speech tagging
word segment/ entity recognition
Return Answer
Vocabulary Set
Training
Bayesian classification method
Classifier
Problem form representation
Query
Knowledge Graph

Fig. 5. Architecture of question answering system.

Focusing on the "people-centered" knowledge graph of poverty alleviation, the training set of poverty alleviation knowledge related issues is constructed. Using Java open source Chinese word segment tool named Ansj (https://github.com/NLPChina/ansj_seg), question training set is segmented and part-of-speech tagged to generate a vocabulary set. User-defined dictionary can be loaded at the same time to make word segmentation more accurate.

Question Text Preprocessing. Ansj is used in word segment. Questions are divided into fine-grained word forms to express, identify names, determine central word of the questions, and ensure that the answers to the questions are obtained for central word.

Abstract Classification of Problems. Naive Bayesian method is simple to implement and has high prediction efficiency. It is widely used in the field of text classification. In order to classify question sentences, we use Bayesian classifier to extract feature vectors of text sentences through training set and One-hot to calculate sentence similarity, so that classifier can effectively deal with natural language question sentences and accurately identify the intention of user questions.

According to Bayesian Classification Calculation to Get Question Categories. According to setting query rules, query sentences corresponding to the categories is executed to match knowledge points from knowledge graph to answer user's questions. Take question "张凤祥的帮扶人是谁 (Who is Zhang Fengxiang's helper?)" as an example, the QA process includes the following four main steps. ① Construct training set of question in poverty alleviation field, and use Ansi toolkit to participle and part-of-speech annotate training set. A vocabulary set of {之子 (son), 之女 (daughter), 配偶 (spouse), 帮扶人 (helper), 帮扶项目 (GovHelp)} is constructed. ② Use Ansj toolkit to participle question text, we can get {"张凤祥","的", "帮扶人", "是", "谁"} ({"Zhang Fengxiang's", "Helper", "is", "who"})and confirm that the central word of the question is "张凤祥 (Zhang Fengxiang)" through person name entity recognition. ③ According to Bayesian classification, the most probabilistic category is the category of the current problem. From the perspective of machine learning, Bayesian formula (Formula 1) interprets X as “having certain characteristics” and Y as “category label” and Bayesian formula can be interpreted as shown in Formula 2.

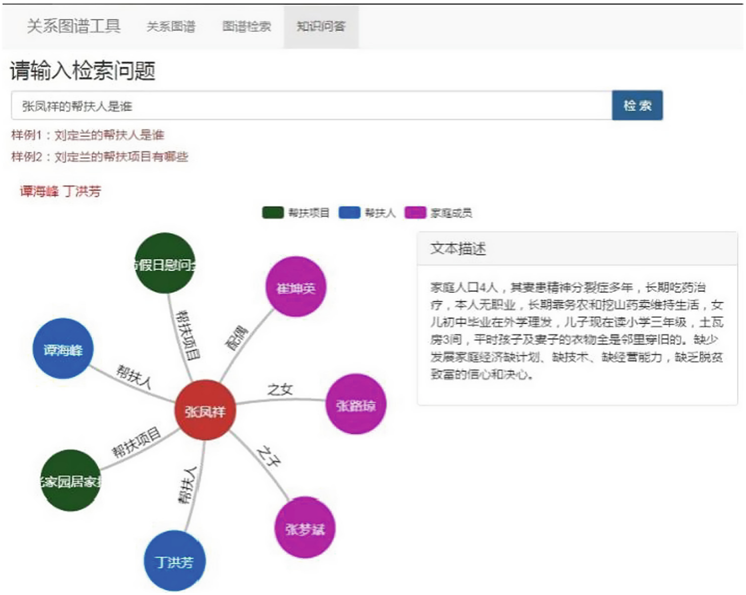
$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{1}$$

$$\begin{aligned} &\mathrm{P}(\text{category label}|\text{having certain characteristics}) \\ &= \frac{\mathrm{P}(\text{having certain characteristics}|\text{category label})\mathrm{P}(\text{category label})}{\mathrm{P}(\text{having certain characteristics})} \end{aligned} \tag{2}$$

According to Formula 2, the probability P (S |(“Who is Zhang Fengxiang’s Helper”) of category S = {Son, Daughter, Spouse, Helper, Support Project} is calculated. As shown in Formula 3, the maximum category of probability value is the category of question.

$$\begin{aligned} &\mathrm{P}(S|(\text{"Who is Zhang Fengxiang's Helper"})) \\ &= \frac{\mathrm{P}(\text{"Who"}|S) * \mathrm{P}(\text{"is"}|S) * \mathrm{P}(\text{"Zhang Fengxiang's"}|S) * \mathrm{P}(\text{"Helper"}|S) * \mathrm{P}(S)}{\mathrm{P}(\text{"Who is Zhang Fengxiang's Helper"})} \end{aligned} \tag{3}$$

According to the calculation, the problem category S = {帮扶人 (Helper)} is obtained, and the knowledge storage (“张凤祥 (Zhang Fengxiang)”, Helper, “谭海峰 (Tan Haifeng))” and (“张凤祥 (Zhang Fengxiang)”, Helper, “丁洪芳 (Ding Hongfang))” in the knowledge map of poverty alleviation is matched and queried. The exact



请输入检索问题

张凤祥的帮扶人是谁

样例1：刘定兰的帮扶人是谁

样例2：刘定兰的帮扶项目有哪些

谭海峰 丁洪芳

文本描述

家庭人口4人，其妻患精神分裂症多年，长期吃药治疗，本人无职业，长期靠务农和挖山药卖维持生活，女儿初中毕业在外学理发，儿子现在读小学三年级，土瓦房3间，平时孩子及妻子的衣物全是邻里穿旧的。缺少发展家庭经济缺计划、缺技术、缺经营能力，缺乏脱贫致富的信心和决心。

Fig. 6. Knowledge question answering query.

answer to the question is {谭海峰 (Tan Haifeng), 丁洪芳 (Ding Hongfang)} through the query sentence. This question is searched in the application system of poverty alleviation knowledge graph as shown in Fig. 6.

6 Conclusion

In this paper, a knowledge graph construction approach is proposed. According to this approach, poverty-alleviation knowledge graph is constructed. This paper analyzes poverty alleviation data to construct poverty alleviation ontology to solve the problem of semantic heterogeneity and provide the upper data model for the construction of poverty-alleviation knowledge graph. Construct Karma models to integrate multi-source data and publish unified RDF data. The RDF2Neo4j interpreter is designed to parse and store RDF data into Neo4j. It provides important data support for big data analysis and government precise poverty alleviation decision-making and provides storage guarantee for the expansion and maintenance of knowledge graph.

Based on Spring Boot framework, visualization technology, natural language processing technology and Bayesian classification algorithm, poverty alleviation knowledge graph application system is designed, which provides users with knowledge graph query and intelligent knowledge question-and-answer functions.

Next, we need to collect poverty alleviation data of different regions, so that the poverty-alleviation knowledge graph application system can provide more accurate reference value in the analysis of poverty alleviation data and the government's precise poverty alleviation decision-making.

Acknowledgment. This work was supported by National Key Research &Development Program of China (No. 2016YFB1001103).

References

1. Juanzi, L.I., Lei, H.O.U.: Reviews on knowledge graph research. J. Shanxi Univ. (Nat. Sci. Ed.) **40**(3), 454–459 (2017)
2. Li, W., Chai, L., Yang, C., Wang, X.: An evolutionary analysis of DBpedia datasets. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 317–329. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_30
3. Ruan, T., Sun, C.-l., Wang, H-f., Fang, Z.-j.: Construction of traditional Chinese medicine knowledge graph and its application. J. Med. Informatics **37**(4), 8–13 (2016)
4. Shijia, E., Lin, P., Xiang, Y.: Automatical construction of Chinese knowledge graph system. J. Comput. Appl. **36**(4), 992–996, 1001 (2016)
5. Yun, H., Xu, J., Knoblock, C.A., Xu, R.: Research and application of multi-source data integration based on ontology. Int. J. u- e- Serv. Sci. Technol. **9**(9), 75–88 (2016)
6. Yun, H.-y., Huang, C., Yu, X.-y., Sui, Y., Hu, G.: Exploiting semantics for conflict event data integration. J. Qingdao Univ. (Nat. Sci. Ed.) **29**(3), 47–52 (2017)
7. Knoblock, C.A., Szekely, P.: Exploiting semantics for big data integration. AI Mag. (S0738-4602) **36**(1), 25–38 (2015)

8. Yun, H., Xu, J., Guo, Z., Wei, X.: Modeling of marine ecology ontology. J. Comput. Appl. **34**(4), 1105–1108 (2015)
9. Szekely, P., et al.: Building and using a knowledge graph to combat human trafficking. In: Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9367, pp. 205–221. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25010-6_12
10. Zhou, Y., Zhou, M., Wang, X., Huang, Y.: Design and implementation of historical fig.s knowledge graph visualization system. J. Syst. Simul. **28**(10), 2560–2566 (2016)